CSC 282 - Fall 2017
http://www.cs.rochester.edu/~stefanko/Teaching/17CS282/

# Name:

Honor Pledge (following http://www.rochester.edu/college/honesty/policy.html#pledge):

I affirm that I will not give or receive any unauthorized help on this exam, and that all work will be my own.

_____

_____

_____

(signature/date)

| |
|---|
| problem 1 |
| problem 2 |
| problem 3 |
| TOTAL |

do not write on this page

1. (40 POINTS) Unlimited supply coin change problem with values and weights. There are $n$ types of coins. We have an unlimited supply of each coin type. The $i$-th coin type has weight $w_i$ and value $v_i$ (where $w_i$ and $v_i$ are both positive integers). We want to pay amount $A$ so that the total weight of the coins used is $B$ and we minimize the total number of coins used. We will solve the problem using dynamic programming. Let $T[a, b]$ be the minimum number of coins that can be used to pay amount $a$ with the total weight of the coins used being $b$ (we let $T[a, b] = \infty$ if it is not possible to pay amount $a$ with the total weight of the coins used being $b$). Give an expression (or a piece of code) to compute $T[a, b]$ from smaller subproblems. You don't have to deal with the base cases. **Clearly explain your expression (or code). Clearly indicate your final answer.**

2. (40 POINTS) A sequence $c_1, \ldots, c_\ell$ is **palindromic** if $c_j = c_{\ell+1-j}$ for all $j \in \{1, \ldots, \ell\}$. We say that a subsequence of a sequence $a_1, \ldots, a_n$ is **valid** if no 2 consecutive numbers of $a_1, \ldots, a_n$ are selected.[1] We are given a sequence $a_1, \ldots, a_n$ and want to find the longest subsequence of $a_1, \ldots, a_n$ that is both valid and palindromic. For $1 \leq i \leq j \leq n$ let $T[i, j]$ be the length of the longest valid palindromic subsequence of $a_i, \ldots, a_j$. Give an expression (or a piece of code) to compute the value of $T[i, j]$ from smaller subproblems. For simplicity assume $j - i \geq 4$. You don't have to deal with the base cases. **Clearly explain your expression (or code). Clearly indicate your final answer.**

[1] Recall that a subsequence of $a_1, \ldots, a_n$ is any sequence of the form $a_{i_1}, \ldots, a_{i_\ell}$ where $\ell \in \{0, \ldots, n\}$ and $1 \leq i_1 < i_2 < \cdots < i_\ell \leq n$. A subsequence is valid if $i_{j+1} - i_j > 1$ for all $j \in \{1, \ldots, \ell - 1\}$.

3. (40 POINTS) A **shuffle** of two strings $A[1..n]$ and $S[1..m]$ is formed by interspersing the characters into a new string, keeping the characters of $A$ and $S$ in the same order (for example, '**sev**e**r**a**l**' is a shuffle of 'seal' and 'evr'). We are given 3 strings $A[1..n], B[1..m], C[1..p]$. We want to check whether there exists a shuffle of $A[1..n]$ and a subsequence[2] of $B[1..m]$ that is $C[1..p]$. We will solve the problem using dynamic programming. Let $T[i, j, k]$ = true if and only if there exists a shuffle of $A[1..i]$ and a subsequence of $B[1..j]$ that is $C[1..k]$. Give an expression (or a piece of code) to compute the value of $T[i, j, k]$ from smaller subproblems. You don't have to deal with the base cases. **Clearly explain your expression. Clearly indicate your final answer.**

[2] A **subsequence** of a string $B[1..m] = B[1]B[2]\dots B[m]$ is a string $B[i_1]B[i_2]\dots, B[i_\ell]$ where $\ell \in \{0, \dots, m\}$ and $1 \le i_1 < i_2 < \cdots < i_\ell \le m$.